

# AUTOMATIC PROGRAMMING TIME SELECTION FOR ONE TIME PROGRAMMABLE MEMORY

Inventors: Tony M. Turner  
Myron Buer

## CROSS REFERENCE TO RELATED APPLICATIONS

- [0001] This application claims priority to U.S. Provisional Patent Application No. 60/449,856, filed on February 27, 2003, entitled AUTOMATIC PROGRAMMING TIME SELECTION FOR ONE TIME PROGRAMMABLE MEMORY, which is incorporated by reference herein.
- [0002] This application incorporates by reference U.S. Patent No. 6,525,955, entitled "Memory Cell With Fuse Element", U.S. Patent Application No. 10/038,021, filed on January 3, 2002, U.S. Patent Application No. 10/041,296, filed on January 8, 2002, and U.S. Patent Application No. 10/197,437, filed on July 18, 2002.

## BACKGROUND OF THE INVENTION

### Field of the Invention

- [0003] The present invention relates to one-time programmable (OTP) memories, and more particularly, to a more efficient system and method for programming the memories faster.

### Related Art

- [0004] The OTP memory has a gate oxide barrier that can be broken down by applying a voltage to one side of the gate. When the barrier is broken down, the resistance to the power rail (or the ground rail) becomes lower and the value is forever stored in the OTP bit. The value is either a 1 or 0.
- [0005] The time required to break down the barrier and program the bit is variable depending on the fabrication process of the silicon. Some memories require a longer programming time to program a bit if the wafer process is skewed to one corner of the fabrication process. Other memories will program in shorter periods of time.

[0006] Thus, a particular problem that exists is the non-uniformity in programming time for one-time programmable memories. For example, a wafer or a particular programmable memory might have 99% of its bits that may be programmed in a fairly short time, for example, 10  $\mu$ sec. However, a handful of bits are such that they require much longer programming, sometimes as long as 1 sec. For a 1K-bit programmable memory, using the lowest common denominator, or in this case, the longest possible time required to program any bit as the time to program all the bits, means that for a 1K-bit memory, 1000 seconds would be required to program it. This means that the throughput of the programming device used to program the memories is extremely low, causing an increase in the cost of manufacturing of the memories.

[0007] Tester time is costly, so a method is needed to determine the minimum amount of programming time to use on the tester.

#### BRIEF SUMMARY OF THE INVENTION

[0008] Accordingly, the present invention is directed to automatic programming time selection for one time programmable memory that substantially obviates, one or more of the disadvantages of the related art.

[0009] There is provided a method of programming a memory including the steps of attempting to program a bit at a designated address for a predetermined time; testing the bit to see if it has been programmed; increasing the predetermined time by approximately an order of magnitude; repeating the previous steps (until the bit at the designated address is programmed; and repeating all the previous steps by advancing the designated address until all bits in the memory are programmed.

[0010] Additional features and advantages of the invention will be set forth in the description which follows, and in part will be apparent from the description, or may be learned by practice of the invention. The advantages of the invention will be realized and attained by the structure particularly pointed

out in the written description and claims hereof as well as the appended drawings.

[0011] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory and are intended to provide further explanation of the invention as claimed.

#### BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

[0012] The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description serve to explain the principles of the invention. In the drawings:

[0013] **FIG. 1** illustrates the proposed programming method according to the present invention.

#### DETAILED DESCRIPTION OF THE INVENTION

[0014] Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

[0015] The programming of a bit in the OTP memory done based on time. If a bit is programmed for time X (e.g., 10  $\mu$ sec), and the tester returns a GOOD status, the programming of the bit is done. If the time X program cycle is complete, and the bit is not programmed correctly yet, the programming can be done again (and again) until it returns a GOOD status. The gate oxide barrier of the OTP memory will have lower and lower resistance with each programming attempt. However lab results show that if a bit fails the first attempt, the subsequent attempts must be longer in duration to reach success.

[0016] The algorithm is implemented in logic that will select the program time based on seeing the same bit get programmed repeatedly. On the first attempt, a default time is selected. On the second attempt, the address of the bit is

compared and found to be the same location, then the programming time is extended by approximately an order of magnitude. On the third attempt, if the address is the same again, the program time is extended even longer. The process can continue.

[0017] After the bit returns a GOOD status, the programming can move on to another bit and the time can drop back to the shorter default value.

[0018] FIG. 1 illustrates a flow diagram of the programming method of the present invention. As shown in FIG. 1, the programming apparatus waits for a program Start command (step 101). Upon the Start command, a bit address is selected for programming (step 102). If the address is equal to the previous address (step 103), program time is increased (step 104). The bit is then programmed (step 105). On the next step, the status of the bit (Good or Bad) is returned to the tester (step 106). The tester then returns back to step 101. If, at step 103, the address is not equal to the previous address, a short (i.e., default or standard) program time is selected (step 107). The bit is then programmed (see step 105, etc.)

[0019] By automatically selecting a longer program time for bits which require it and automatically selecting a short time for those bits that program quickly, tester time is saved, and no user intervention is needed. This algorithm also applies to “in system” programming as well, where time is not important, but success of programming is important.

[0020] As an alternative, a set of bits can be used to determine the wafer process. These bits are programmed once for a short period. If any bits fail, then the wafer is assumed to be slow to program and a configuration bit (or bits) in the OTP is then programmed that will cause all future program times to be extended by roughly an order of magnitude.

[0021] Thus, the default starting point for the program time can be selected based on the wafer process.

[0022] For example, if a “slow to program” part is found during the test bit programming, this part would then have all its bits programmed in the OTP

memory to have programming times longer than a "short to program" part. This will be true for tester as well as in system programming.

[0023] The programming method of the present invention tests each bit to see whether it belongs to a fast programmable bit or a slow programmable bit. Most bits are typically of the fast programming type, and may be programmed, for example, in 10-20  $\mu$ sec. In that case, the method involves programming a bit that corresponds to a given address, and then testing to see if the bit has in fact been programmed during the 10  $\mu$ sec. If the bit programming was successful, then the programming apparatus moves onto the bit at the next address. If the programming has not been successful, then the tester increases the time required to program that particular bit by, for example, a factor of 10. The tester then tries again to program the bit. Following the next attempt, the tester again checks to see if the bit has been successfully programmed. If it has, then it moves onto the bit at the next address. If it has not, then it again increases the time required to program this particular bit by, for example, a factor of 10, and the process repeats itself.

[0024] It will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined in the appended claims. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.